[Tyler Olivieri]

ECE 3522: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

Variance can be thought of as how much a value can stray away from the mean. It can also be thought of as the energy of the signal. In certain situations keeping the variance closer to the expected value is needed or not to have large changes in the variance. The variance is to be calculated with three different methods of both the audio signal and google stock data. The three methods are calculating the variance of the entire data set, calculate the variance using a growing window or calculating the variance every time a new point is read serially, and calculating the variance using a frame and window approach. This will help observe different methods of estimating variance of a time varying signal.

# Approach and Results

To see the variance in three different views frame and window processing, a growing window used to simulate streaming data, and the variance of the entire data set will be used. The frame and windows processing method will be the same as previous assignments. The variance of the entire data set will be calculated with the MATLAB function, variance. The growing window method will be calculated using a for loop, an index N, and the MATLAB function variance.

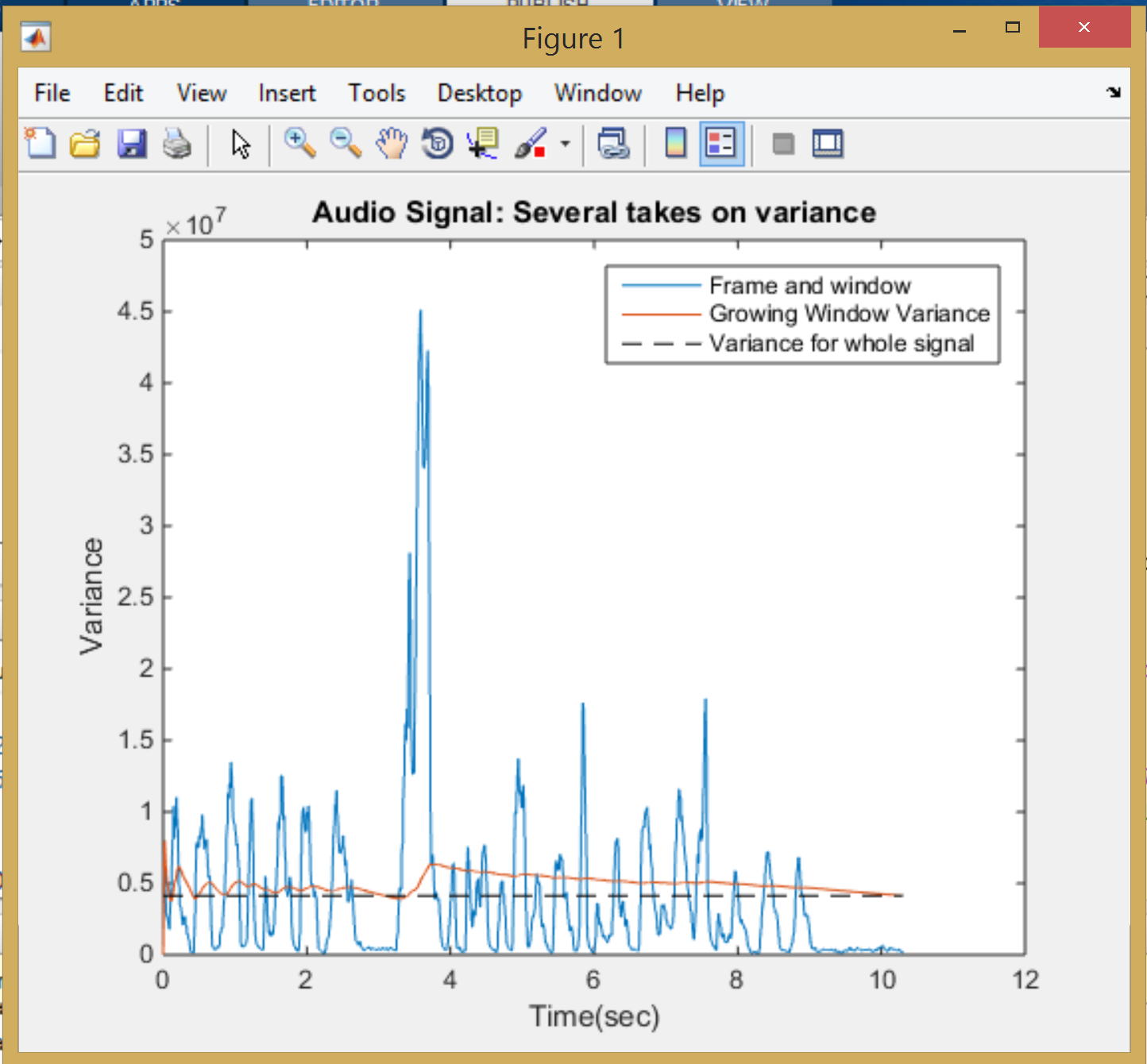


Figure : Audio signal variance shown in 3 different ways

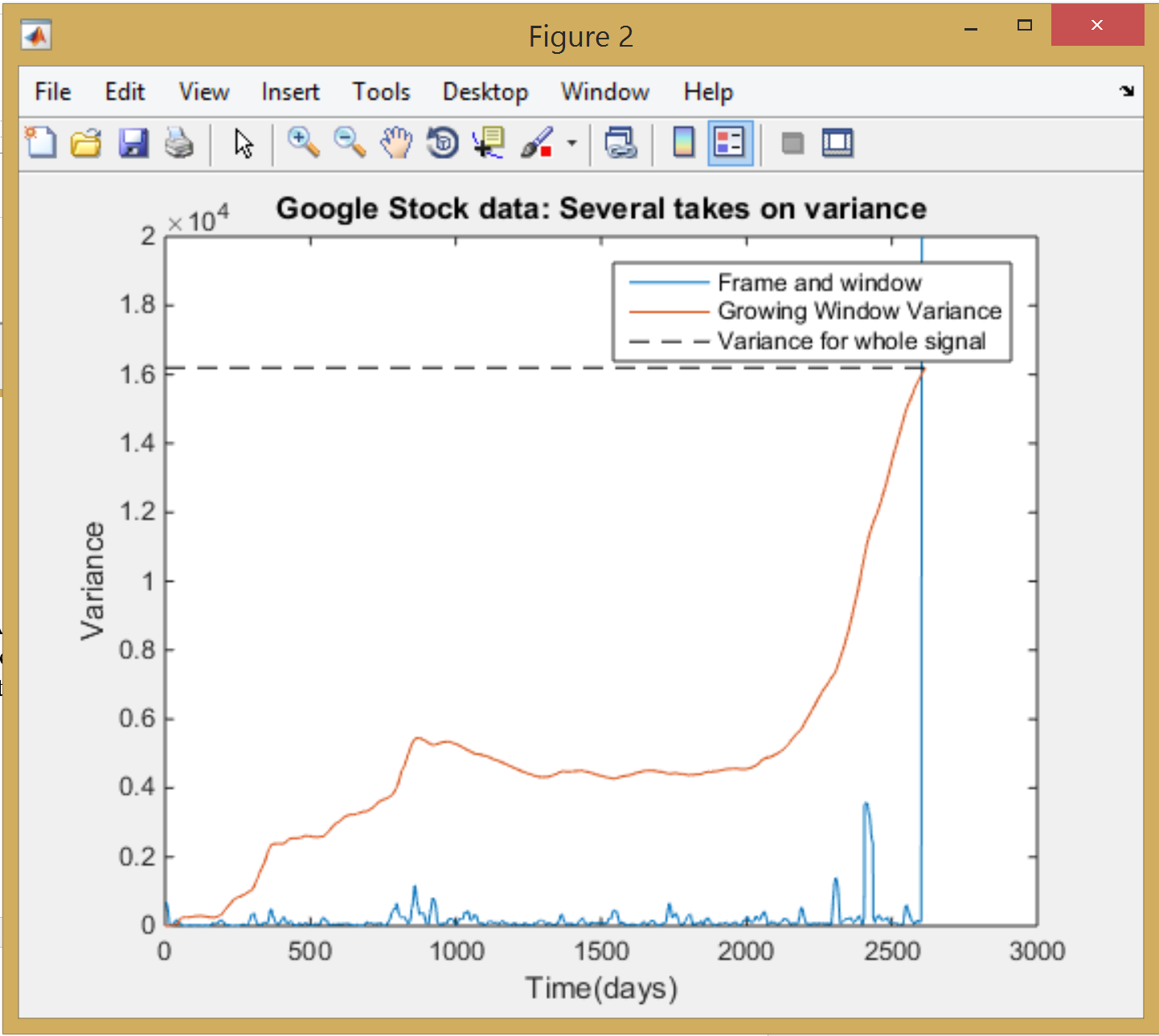


Figure : Very beginning of the audio signal

Above in figure 1, the audio signal variance can be observed. As N increases the variance of the signal will approach the variance of the entire data set. The last points will match because when N is the last point it is taking the variance of the whole data set. The frame and window variance oscillates around the expected variance value. There can be more variance when taking the frame and window of an audio signal. Even with a smaller data set (a window) the variance can be high as the amplitude will drop from a possible large positive peak to a large negative peak with the average of that window being around 0. This is the nature of an oscillating motion like sound. The variance when calculated with the growing window will approach the variance of the entire data set as the growing window gets closer and closer to being the whole data set. The frame and window can show short term variance spikes in the data. When the frame and window method shows variance spikes it can also be seen that the variance of the growing window also spikes. The bigger the spike in the frame and window method, the larger the spike in the growing window method. The growing window then moves to the expected variance value until it reaches then end of the set and the last N value is the same as the expected variance value.

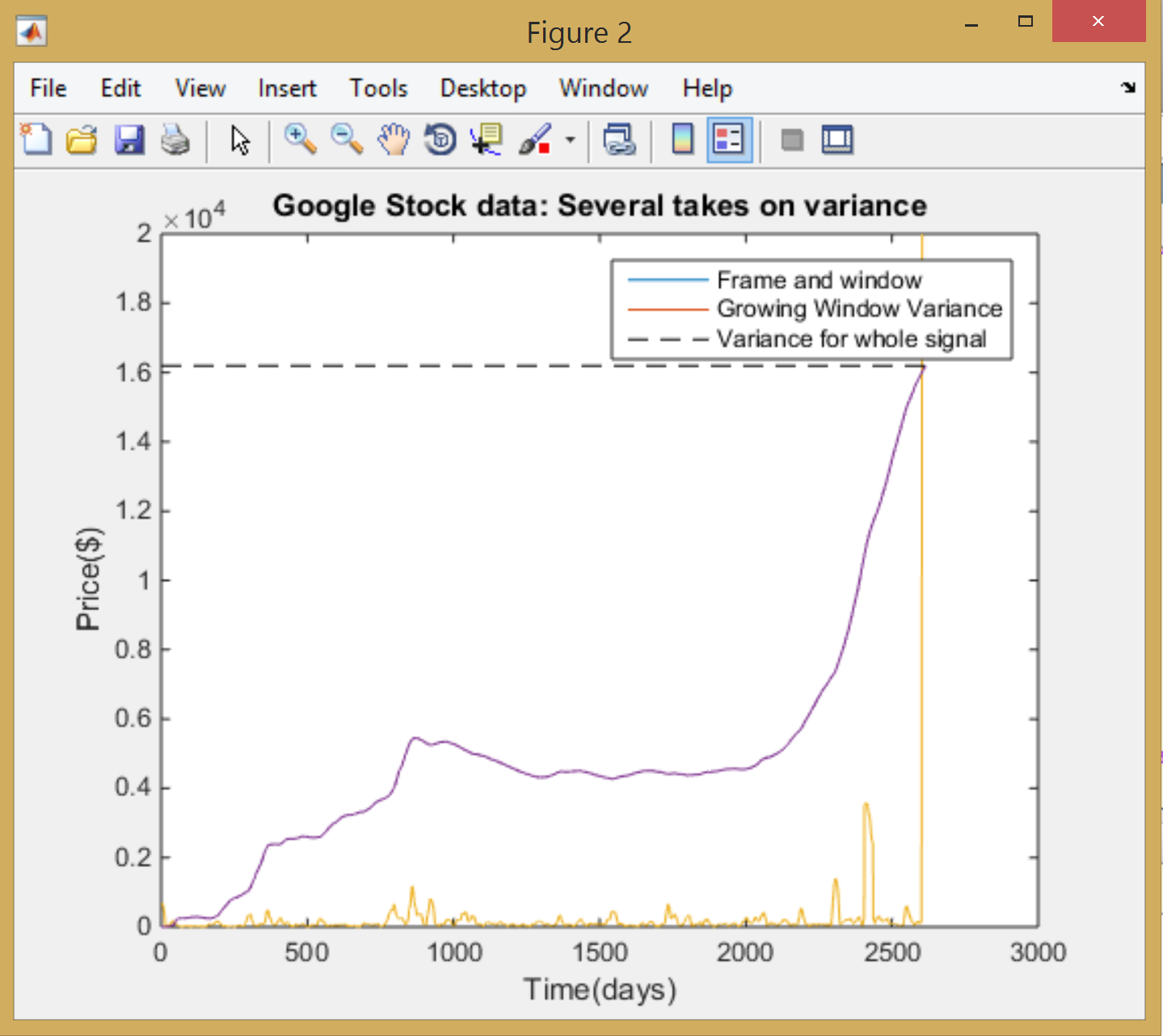


Figure : Google closing stock data variance shown in 3 different ways

In the google stock data, similar things occur, the short term variance spikes will show increases in the variance of the entire data set. The frame and window variance is also much smaller because the google stock did not change drastically day to day or week to week. The google stock data consistently increases with dropping too much at any point. This means that the dataset is getting larger and there are more different values, increasing the variance. This shows why the variance in the growing window method is constantly increasing up to the expected variance for the whole set.

# MATLAB Code

First, the data has to be read in. The closing data also has to be separated from the google stock matrix. Then the variance will be calculated for each entire data set. This will be saved for later. Next, the growing window method will be used on both signals and stored in a matrix for later. This is accomplished by having an index N that starts from 0 and goes to the end of the signal. At every N value a temporary matrix is created to go from (1:N) of the closing stock data. The variance is then calculated for this temporary matrix and stored in matrix with index N. This will eventually be as long as the original signal and then can be plotted against the same time vector. Then the frame and window method is used to process the data with the frame and window combinations given. Now that all the data is in matrix we can plot them. Potting the variance of the entire set is accomplished with the line function. A line tht looks like ([0, length of time vector] [variance of the set, variance of the set]). Then using hold on, the rest of the plots can be plotted and titled.

clc;clear;clf;  
%----Read in data-----  
stock\_data = xlsread('google\_v00.xlsx');  
fp =fopen('rec\_01\_speech.raw');  
audio\_signal = fread(fp,inf, 'int16');  
fclose(fp);  
  
%----Separate close so that it can be analyzed  
  
close =stock\_data(:,4);  
  
%----variance  
variance\_s = var(close);  
variance\_a = var(audio\_signal);  
  
%variance from the first N samples of closing data,  
%letting N vary from 0 to a maximum of the number of samples in the file  
for N= 1:length(close)  
 tempN\_close= close(1:N);  
 Var\_closeN(N) = var(tempN\_close);  
end  
  
%variance from the first N samples of audio signal,  
%letting N vary from 0 to a maximum of the number of samples in the file  
for N= 1:length(audio\_signal)  
 tempN\_audio= audio\_signal(1:N);  
 Var\_audioN(N) = var(tempN\_audio);  
end  
  
%----- stock\_data  
sig\_a = stock\_data;  
fdur\_a = 1;  
wdur\_a = 30;  
sig\_wbuf = zeros(1, wdur\_a);  
num\_samples = length(sig\_a);  
num\_frames = 1+round(num\_samples / fdur\_a);  
mean\_stock = zeros(length(sig\_a),1);  
  
% loop over the entire signal  
%  
 for i = 1:num\_frames  
  
 % generate the pointers for how we will move through the data signal.  
 % the center tells us where our frame is located and the ptr and right  
 % indicate the reach of our window around that frame  
 %  
 n\_center = (i - 1) \* fdur\_a + (fdur\_a / 2);  
 n\_left = n\_center - (wdur\_a / 2);  
 n\_right = n\_left + wdur\_a ;  
  
 n\_right = round(n\_right);  
 n\_left = round(n\_left);  
 % when the pointers exceed the index of the input data we won't be  
 % adding enough samples to fill the full window. to solve this zero  
 % stuffing will occur to ensure the buffer is always full of the same  
 % number of samples  
 %  
 if( (n\_left < 0) || (n\_right > num\_samples) )  
 sig\_wbuf = zeros(1, wdur\_a);  
 end  
  
 % transfer the data to this buffer:  
 % note that this is really expensive computationally  
 %  
 for j = 1:wdur\_a  
 index = n\_left + (j - 1);  
 if ((index > 0) && (index <= num\_samples))  
 sig\_wbuf(j) = sig\_a(index);  
 end  
 end  
  
 %calculate mean and variance for  
 %current window  
  
 var\_stock = var(sig\_wbuf);  
  
 % assign the mean/variance value to the output signal:  
 % note that we write fdur\_a values  
 %  
 for j = 1:fdur\_a  
 index = n\_center + (j - 1) - (fdur\_a/2);  
 if ((index > 0) && (index <= num\_samples))  
 var\_stock\_full(index) = var\_stock;  
 end  
 end  
  
 end  
  
% --audio signal frame: 10ms window: 30ms  
sig\_a = audio\_signal;  
fdur\_a = 80;  
wdur\_a = 240;  
sig\_wbuf = zeros(1, wdur\_a);  
num\_samples = length(sig\_a);  
num\_frames = 1+round(num\_samples / fdur\_a);  
mean\_audio = zeros(length(sig\_a),1);  
  
% loop over the entire signal  
%  
 for i = 1:num\_frames  
  
 % generate the pointers for how we will move through the data signal.  
 % the center tells us where our frame is located and the ptr and right  
 % indicate the reach of our window around that frame  
 %  
 n\_center = (i - 1) \* fdur\_a + (fdur\_a / 2);  
 n\_left = n\_center - (wdur\_a / 2);  
 n\_right = n\_left + wdur\_a ;  
  
 n\_right = round(n\_right);  
 n\_left = round(n\_left);  
 % when the pointers exceed the index of the input data we won't be  
 % adding enough samples to fill the full window. to solve this zero  
 % stuffing will occur to ensure the buffer is always full of the same  
 % number of samples  
 %  
 if( (n\_left < 0) || (n\_right > num\_samples) )  
 sig\_wbuf = zeros(1, wdur\_a);  
 end  
  
 % transfer the data to this buffer:  
 % note that this is really expensive computationally  
 %  
 for j = 1:wdur\_a  
 index = n\_left + (j - 1);  
 if ((index > 0) && (index <= num\_samples))  
 sig\_wbuf(j) = sig\_a(index);  
 end  
 end  
  
 %calculate mean and variance for  
 %current window  
 var\_audio = var(sig\_wbuf);  
  
 % assign the mean/variance value to the output signal:  
 % note that we write fdur\_a values  
 %  
 for j = 1:fdur\_a  
 index = n\_center + (j - 1) - (fdur\_a/2);  
 if ((index > 0) && (index <= num\_samples))  
 var\_audio\_full(index) = var\_audio;  
 end  
 end  
  
 end  
  
%---- time vector stock  
t\_stock = linspace(0, length(stock\_data),length(stock\_data));  
  
%----time vector audio signal  
samples = length(audio\_signal);  
fs = 8000;  
sampleperiod = 1/fs;  
Length = samples\*sampleperiod;  
t\_audio = linspace(0, Length, samples);  
  
  
%---- plot of variance(constant) audio signal  
%----w/ frame window analysis  
  
figure(1);  
plot(t\_audio, var\_audio\_full)  
hold on  
plot(t\_audio, Var\_audioN)  
j = line([0 Length], [variance\_a variance\_a]);  
j.LineStyle ='--';  
j.Color= [0 0 0];  
title('Audio Signal: Several takes on variance');  
xlabel('Time(sec)');  
ylabel('Amplitude');  
legend('Frame and window','Growing Window Variance','Variance for whole signal');  
%---- plot of variance (constant) closing price  
%---- w/ frame window analysis  
  
figure(2);  
plot(t\_stock, var\_stock\_full)  
hold on  
plot(t\_stock, Var\_closeN)  
l =line([0 length(t\_stock)], [variance\_s variance\_s])  
l.LineStyle ='--';  
l.Color = [ 0 0 0];  
title('Google Stock data: Several takes on variance');  
xlabel('Time(days)');  
ylabel('Price($)');  
legend('Frame and window','Growing Window Variance','Variance for whole signal');

# Conclusions

Variance corresponds to changes in the signal. Different approaches to calculate variance can show the variance in different lights. Calculating the variance with a frame and window method with show the variance in short term. While calculating the variance by the growing method will show the variance in more of a long term fashion. In terms of streaming an audio signal, calculating the variance using the growing window method would be better to keeping the volume more steady instead of large possible spikes using frame and windows. While with the google stock data the frame and window variance could be useful for people betting to see if the price has varied a lot recently.